The Examiner requires that Fig. 1 be designated with the legend --Prior Art--. A proposed drawing correction incorporating such a change is submitted herewith for the Examiner's approval.

The Examiner rejected claims 1, 2, 4, 32-36, 38, and 66-69 under 35 USC §102(b) as being anticipated by Hirsch.

Independent claim 1 recites a cryptographic key split combiner. The combiner includes a plurality of key split generators for generating cryptographic key splits, and a key split randomizer for randomizing the cryptographic key splits to produce a cryptographic key. Each of the key split generators includes means for generating key splits from seed data.

In contrast, Hirsch discloses a software data protection mechanism, which includes an apparatus for generating a key. The description of the key generating apparatus is set forth in detail from column 3, line 47 through column 5, line 28, with reference to Figs. 1A-C. First, a 32-bit binary value is loaded into an input register 12. Each bit of the 32-bit value is provided as an input to a corresponding scrambler array container 140-n, n=0-31. Each container 140-n in the scrambler array 14 determines whether the input bit of that container is passed to a respective bit position of a 32-bit output register 18, or whether the complement of that bit is passed instead.

The control for determining whether the input bit or its complement is passed is determined by every fifth bit of a pseudorandom sequence that is generated by a pseudorandom number generator 16. This generator takes a single seed and generates a single sequence that is loaded serially into the containers of the scrambler array 14, such that five bits are stored in each container. Because there are 32 containers, a 5-bit number also designates each bit position (00000, 00001, 00010, 00011, … , 11101, 11110, 11111). The result of an XOR operation on the LSB of the bit position and the LSB of the 5-bit pseudorandom sequence segment stored in the container determines whether the input bit or its complement will be provided to the output register. Because the LSB of the bit position alternates in the sequence of scrambler array containers, the XOR result for all even-numbered containers will be the pseudorandom number LSB, and the XOR result for all odd-numbered containers will be the pseudorandom number LSB complement. The actual bit position of the output register to which the respective "scrambled" input value bits is provided is determined by the actual value of the pseudorandom number, although the exact correspondence is not disclosed by Hirsch.

The 32-bit value stored in the output register 18 is now encoded by mapping the values of grouped bits according to a table. That is, the 32 bits of output data are separated into four groups of eight bits each. Each 8-bit group is stored in a set of two overlapping 5-bit registers 200-1a, 200-1b. The outputs of these registers (eight 5-bit values) are provided to an alphanumeric table 204, which maps the input values to provide eight alphanumeric values. The sequence consisting of these values is the key, which is stored in a user key register 24.

Thus, where claim 1 recites a plurality of key split generators, each receiving seed data and generating a respective key split, and a randomizer for receiving and randomizing the plurality of key splits to generate a key, Hirsch discloses an array 14 for scrambling a single value according to a single pseudorandom sequence generated from a single seed, and an encoder 20 for mapping the scrambled value to generate the key. That is, Hirsch does not disclose a plurality of key split generators, each receiving seed data and generating a respective key split, as recited in claim 1. Rather, Hirsch discloses a plurality of containers in a single scrambler array, which together receive a single serially-shifted pseudorandom sequence generated from a single seed. Further, Hirsch does not disclose randomizing a plurality of separately generated key splits to generate a key, as recited in claim 1. Rather, Hirsch discloses taking a single 32-bit scrambled value, and mapping 8-bit segments of that value according to a stored, predetermined table, to generate a key.

In summary, the combiner of claim 1 takes separate key splits, generated based on separate seed values, and randomizes the splits to provide a key. In contrast, Hirsch takes a single input value, scrambles the value according to a single pseudorandom stream based on a single seed, and maps the scrambled value according to a fixed table to generate the key. It is clear that the Hirsch apparatus is quite different from the combiner recited in claim 1, and does not include any of the elements recited in claim 1. Thus, Hirsch cannot anticipate the invention recited in claim 1, nor that recited in claims 2, 4, and 32-34, which all depend from claim 1. The rejection of these claims, therefore, should be withdrawn.

Claim 35 recites a process for forming cryptographic keys. The process includes generating a plurality of cryptographic key splits from seed data; and randomizing the cryptographic key splits to produce a cryptographic key.

As noted above, Hirsch does not disclose this process. Hirsch does not generate a plurality of cryptographic key splits; Hirsch generates a single scrambled value from a single 32-bit value. Hirsch does not generate key splits from seed data. Hirsch only generates a pseudorandom sequence from a seed; the sequence is used to scramble the input value, not to generate a split on which the key is based. Hirsch does not disclose randomizing key splits to produce a cryptographic key; Hirsch discloses mapping a single scrambled value to generate a key.

Thus, Hirsch does not disclose the process recited in claim 35, nor that recited in claims 36 and 38, which depend from claim 35. The rejection of these claims, therefore, should be withdrawn.

Claim 66 recites a cryptographic key formed by the process of claim 35. Hirsch discloses a different process, and a different apparatus for carrying out the key-generating process, and therefore cannot disclose the recited key. Hirsch, therefore, cannot anticipate the invention recited in claim 66, nor that recited in claims 67-69, which depend from claim 66. The rejection of these claims, therefore, should be withdrawn.

The Examiner rejected claims 3 and 37 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Albert et al. Claim 3 recites the combiner of claim 1, wherein the plurality of key split generators includes a random split generator for generating a random split based on reference data, and the random split generator includes means for generating a random sequence based on the reference data.

As noted above, Hirsch does not disclose the combiner recited in claim 1. Albert et al. merely disclose a random number generator. There is no suggestion by Albert et al. that the disclosed random number generator should or could be used as part of a random split generator in a cryptographic key split combiner as recited in claim 1. Further, Albert et al. does not overcome the deficiencies of Hirsch in disclosing the combiner recited in claim 1. That is, Hirsch does not disclose a combiner that takes separate key splits, generated based on separate seed values, and randomizes the splits to provide a key, as recited in claim 1. Albert et al., by merely describing a random number generator, do not provide Hirsch with a means for generating separate key splits from separate seed values, or for randomizing separate splits to provide a key.

The Examiner does not specify how the teachings of these two references could be combined, but implied that the Albert et al. random number generator could be substituted for the Hirsch pseudorandom number generator to provide less predictability. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component.

For at least all the above-noted reasons, no combination of the teachings of the cited references could render obvious the invention recited in claim 3. The rejection of claim 3, therefore, should be withdrawn.

Claim 37 recites the process of claim 35, wherein generating a plurality of cryptographic key splits includes generating a random key split based on reference data, and wherein generating a random key split includes generating a random sequence based on the reference data

As noted above, Hirsch does not disclose the process recited in claim 35. Albert et al. merely disclose a random number generator. There is no suggestion by Albert et al. that the disclosed random number generator should or could be used to generate a random key split in the process recited in claim 35. Further, Albert et al. does not overcome the deficiencies of Hirsch in disclosing the process recited in claim 35. That is, Hirsch does not disclose generating a plurality of cryptographic key splits, generating key splits from seed data, or randomizing key splits to produce a cryptographic key, as recited in claim 35. Hirsch generates a single scrambled value from a single 32-bit value, only generates a single pseudorandom sequence from a single seed, only uses the sequence to scramble the input value rather than to generate a split on which the key is based, and maps a single scrambled value to generate a key. Albert et al., by merely describing a random number generator, do not provide Hirsch with a means for generating separate key splits from separate seed values, or for randomizing separate splits to provide a key.

The Examiner does not specify how the teachings of these two references could be combined, but implied that the Albert et al. random number generator could be substituted for the Hirsch pseudorandom number generator to provide less predictability. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component.

For at least all the above-noted reasons, no combination of the teachings of the cited references could render obvious the invention recited in claim 37. The rejection of claim 37, therefore, should be withdrawn.

The Examiner rejected claims 5 and 39 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Thomlinson et al.

Claim 5 depends from claim 1 and recites that the random split generator includes means for generating a key split based on the reference data and on chronological data.

Thomlinson et al. disclose a non-biased pseudorandom number generator. The generator includes an input device that gathers classes of bits provided elsewhere in a computer to hash a seed used to generate the number. One of the classes of bits is a machine class that relates to operating parameters of the computer, such as time of day and date.

As noted above, Hirsch does not disclose the combiner recited in claim 1. Thomlinson et al. merely disclose a pseudorandom number generator. There is no suggestion by Thomlinson et al. that the disclosed number generator should or could be used as part of a random split generator in a cryptographic key split combiner as recited in claim 1. Further, the Thomlinson et al. reference does not overcome the deficiencies of Hirsch in disclosing the combiner recited in claim 1. That is, Hirsch does not disclose a combiner that takes separate key splits, generated based on separate seed values, and randomizes the splits to provide a key, as recited in claim 1. Thomlinson et al., by merely describing a pseudorandom number generator, do not provide Hirsch with a means for generating separate key splits from separate seed values, or for randomizing separate splits to provide a key.

The Examiner does not specify how the teachings of these two references could be combined, but implied that the Thomlinson et al. pseudorandom number generator could be substituted for the Hirsch pseudorandom number generator, because the time and date component of the number would make the number harder to guess. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component. Thus, substituting the Thomlinson et al. pseudo random number for the Hirsch pseudorandom number is simply replacing one number that is not a key split for another number that is not a key split. As a result, the limitations of claims 1 and 5 are still not satisfied.

For at least all the above-noted reasons, no combination of the teachings of the cited references could render obvious the invention recited in claim 5. The rejection of claim 5, therefore, should be withdrawn.

Claim 39 depends from independent claim 35, and recites that generating a random key split includes generating a key split based on the reference data and on chronological data.

As noted above, Hirsch does not disclose the process recited in claim 35. Thomlinson et al. merely disclose a pseudorandom number generator. There is no suggestion by Thomlinson et al. that the disclosed random number generator should or could be used to generate a random key split in the process recited in claim 35. Further, the Thomlinson et al. reference does not overcome the deficiencies of Hirsch in disclosing the process recited in claim 35. That is, Hirsch does not disclose generating a plurality of cryptographic key splits, generating key splits from seed data, or randomizing key splits to produce a cryptographic key, as recited in claim 35. Hirsch generates a single scrambled value from a single 32-bit value, only generates a single pseudorandom sequence from a single seed, only uses the sequence to scramble the input value rather than to generate a split on which the key is based, and maps a single scrambled value to generate a key. Thomlinson et al., by merely describing a pseudorandom number generator, do not provide Hirsch with a means for generating separate key splits from separate seed values, or for randomizing separate splits to provide a key, as recited in claim 35.

The Examiner does not specify how the teachings of these two references could be combined, but implied that the Thomlinson et al. pseudorandom number generator could be substituted for the Hirsch pseudorandom number generator, because the time and date component of the number would make the number harder to guess. However, the Hirsch pseudorandom number generator does not provide a key split, that is, a component of the generated key. Rather, this generator provides one element used to scramble the single 32-bit input in the scrambler array that is the only component of the generated key. It is a functional input, rather than a key component. Thus, substituting the Thomlinson et al. pseudo random number for the Hirsch pseudorandom number is simply replacing one number that is not a key split for another number that is not a key split. As a result, the limitations of claims 35 and 39 are still not satisfied.

For at least all the above-noted reasons, no combination of the teachings of the cited references could render obvious the invention recited in claim 39. The rejection of claim 39, therefore, should be withdrawn.

The Examiner rejected claims 6, 7, 9-11, 13-16, 40, 41, 43-45, and 47-50 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Ming et al.

Claims 6, 7, 9-11, and 13-16 all depend from claim 1, and claims 40, 41, 43-45, and 47-50 all depend from claim 35. As thoroughly discussed above, Hirsch does not disclose the invention recited in claims 1 and 35. Ming et al. disclose encoder and decoder apparatus for a cable television signal having embedded viewer access control data. The Ming et al. invention provides a means for blocking selected television programming classes in a cable television transmission, based on access privileges of a viewer at a television receiver having a decoder unit. Access denial codes are embedded in the transmitted video signal, and are read by the decoding unit, which implements the blocking function. Programs that the viewer is not authorized to access are scrambled, wherein the scrambling function takes the form of random line inversion. Thus, Ming et al. do not overcome the deficiencies of Hirsch, that is, Ming et al. do not even disclose a key split combiner. This is reason enough to conclude that no combination of the teachings of Hirsch and Ming et al. could render obvious the invention recited in claims 6, 7, 9-11, 13-16, 40, 41, 43-45, and 47-50. However, the features recited in these dependent claims even further distinguish the claimed invention from the cited references.

With regard to claims 6 and 40, the Examiner stated that Ming et al. disclose generating a key split based on static data, at column 4, lines 4-7. That passage describes generating and storing an initial seed value. This seed value is used in a pseudorandom sequence generator as the random function driving the line inverter. See column 14, lines 30- column 15, line 15. Thus, generating and storing the seed value by Ming et al. is not the same as generating a static key split. Ming et al. have no key; the invention simply performs random line inversion. Even if the pseudorandom sequence were considered a key, it has no separate components, that is, no splits. Neither Hirsch nor Ming et al. discloses a randomizing key split combiner that generates a cryptographic key from a number of splits, which are generated from separate seeds, and no combination of the teachings of these references could render obvious the invention recited in claims 6 and 40. The rejection of these claims should be withdrawn.

Regarding claims 7 and 41, the Examiner stated that Ming et al. disclose a means of updating the static data, at column 4, line 8. That passage describes generating a next seed value. This seed value is used to generate a new pseudorandom sequence for the subsequent video frame. As noted above in discussing the rejection of claims 6 and 40, this is not an update of static data used in generating a key split. The rejection of claims 7 and 41 should be withdrawn.

Regarding claims 9 and 43, the Examiner stated that Ming et al. disclose a token split generator for generating a token key split based on label data, at column 6, lines 26-29; column 5, lines 65-67; and column 6, lines 1-5. In those passages, Ming et al. disclose the use of five levels of viewer access. However, these are not key splits, and they have nothing to do with tokens. The access control data described by Ming et al. is embedded in the video data; it is not used to generate a key split. See column 13, lines 16-20. Further, this data is not derived from a token. It is clear that no combination of the teachings of the cited references could render obvious the invention recited in claims 9 and 43. The rejection of these claims should be withdrawn.

Regarding claims 10 and 44, the Examiner stated that Ming et al. suggest reading label data from a storage medium, at column 7, lines 11-22. That passage describes that the decoder apparatus has a prestored user address that corresponds to a user address in the access control data. First, because the user address is stored at the decoder and compared with incoming address data, it is not the basis for a key split that is generated and then randomized with other key splits. Further, the fact that it is prestored at the decoder apparatus demonstrates that this data could not be the basis for a token key split, even if it were part of some key split, and even if Ming et al. used any key splits at all. The rejection of claims 10 and 44 should be withdrawn.

Regarding claims 11 and 45, the Examiner stated that Ming et al. describe label data includes user authorization data, at column 7, lines 22-25. That passage describes that the video data includes a program class code identifying authorized classes of users. However, this program class code is not the basis for key split data; it is merely embedded in the video data, and is not a component of a cryptographic key. Further, even if it were a key split, there is no mention of a token, so it could not be the basis of a token key split. The rejection of claims 11 and 45 should be withdrawn.

Regarding claims 13 and 47, the Examiner stated that Ming et al. illustrate a means for generating a pseudorandom sequence, based on label data, at column 13, lines 45-50; Figure 2, items 113-115; and column 14, lines 39-44. In those passages, Ming et al. disclose two different processes. The column 13 and Figure 2 references describe how access control data from two different channel processors are alternately encrypted using the conventional DES algorithm. Thus, the data that the Examiner refers to as the label data is encrypted, but no pseudorandom sequence is generated. The column 14 passage describes the process taking place in Figure 3, which in part shows details of the data formatter and video scramble control block 118 in Figure 2. This block receives an 8-bit value 121 from an unknown source, and a line signal 122. Based on these signals, a pseudorandom sequence is generated by the generator 120. This sequence is used to drive the random line inverter. Thus, what the Examiner considers to be label data is encrypted, and data from a completely different source is used to generate a pseudorandom sequence. Thus, the "label data" and the pseudorandom sequence are completely unrelated. Further, neither of these has anything to do with a key or with key splits. The "label data" is encrypted for secure transmission to the decoder apparatus, and the sequence is used to directly scramble the substantive data of the transmission in a manner that is unrelated to the access control data. Again, neither process involves any token, let alone a token split. Clearly, the requirements of claims 13 and 47 are not satisfied. The rejection of these claims should be withdrawn.

Regarding claims 14 and 48, the Examiner stated that Ming et al. specify the means for generating a key split based on label data and organization data, at column 6, lines 26-29 and 59-65; column 5, lines 65-67; and column 6, lines 1-5. In those passages, Ming et al. disclose the use of five levels of viewer access, based on different categories particular to users and classes of users. However, these are not key splits, and they have nothing to do with tokens. The access control data described by Ming et al. is embedded in the video data; it is not used to generate a key split. See column 13, lines 16-20. This data, after being decrypted at the decoder, is the subject of a direct comparison to determine if access is granted. The data is not the basis for generating a split that will be randomized to form a key that will be used in the encryption process. Further, this data is not derived from a token. It is clear that no combination of the teachings of the cited references could render obvious the invention recited in claims 14 and 48. The rejection of these claims should be withdrawn.

Regarding claims 15 and 49, the Examiner stated that Ming et al. suggest a means for generating a key split based on the label data and on static data, at column 4, lines 4-7. That passage describes generating and storing an initial seed value. This seed value is used in a pseudorandom sequence generator as the random function driving the line inverter. See column 14, lines 30- column 15, line 15. Thus, generating and storing the seed value by Ming et al. is not the same as generating a static key split. Ming et al. have no key; the invention simply performs random line inversion. Even if the pseudorandom sequence were considered a key, it has no separate components, that is, no splits. Neither Hirsch nor Ming et al. discloses a randomizing key split combiner that generates a cryptographic key from a number of splits, which are generated from separate seeds, and no combination of the teachings of these references could render obvious the invention recited in claims 15 and 49. The rejection of these claims should be withdrawn.

Regarding claims 16 and 50, the Examiner stated that Ming et al. disclose a means of updating the static data, at column 3, lines 65-67 and column 4, lines 1-4. This passage describes several differentiated layers used to determine access by a user to the cable television transmission, what the Examiner had previously referred to when discussing label data. This is not static data on which a key split is based as generated by a token key split generator. Further, it is not disclosed that this data is updated. As shown in Figure 2, this data is encrypted and transmitted to the decoder apparatus. It is not used to generate a key split that is randomized with other key splits to generate a cryptographic key. The rejection of claims 16 and 50 should be withdrawn.

In summary, both the Hirsch invention and the Ming et al. invention are so different from the claimed invention that no combination of the teachings of these two references in an attempt to result in the claimed invention is suggested or possible.

The Examiner rejected claims 8 and 42 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Anshel et al.

The Examiner stated that Ming et al., at column 4, lines 18-20, describe modifying the divisor of static data. However, a close reading of this passage does not reveal the subject matter described by the Examiner. Rather, the passage sets forth a procedure for updating seed values for generating the pseudorandom sequence, based on an incremented frame count. Anshel et al. disclose a cryptographic sequence generator. In the passage cited by the Examiner, Anshel et al. describe use of randomly-selected prime numbers to generate a list of Jacobi symbols and, according to a public key, chooses a subset of the symbols to encrypt a single input bit. The Jacobi symbols and the public key both feature terms having prime number divisors. See column 11, lines 8-53. In contrast, claims 8 and 42 recite updating the static data by modifying a prime number divisor of the static data. The static data is a basis for a random key split.

None of the cited references discloses random key splits. It follows that none of the cited references discloses updating a basis of a random key split, or performing the update by modifying a prime number divisor of the basis. Anshel et al. disclose the use of prime number divisors in a Jacobi sequence and in a public key. However, demonstrating that it is known to use prime number divisors in an application related to cryptography is not enough to suggest to one of ordinary skill in the art that modification of a prime number divisor of a basis of a random key split is beneficial in generating a key based on the randomization of a number of key splits, particularly in view of other references that do not even disclose the use of key splits. It is clear that no combination of the teachings of the cited references could possibly render obvious the invention recited in claims 8 and 42. The rejection of these claims, therefore, should be withdrawn.

The Examiner rejected claims 12 and 46 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Albert et al.

As noted above, Hirsch does not disclose the invention recited in claims 1 and 25. The Examiner asserted that Ming et al. disclose generating a pseudorandom sequence based on label data. However, as noted in discussing the rejection of claims 13 and 47, this is not the case. The access control data is encrypted for transmission to the decoder apparatus, and different data is used to generate a pseudorandom sequence to drive a line inverter. Further, Albert et al. merely disclose a design for a random number generator. Albert et al. do not disclose or suggest anything that would lead one of ordinary skill in the art, given the other references, to develop a cryptographic key split combiner as recited in claims 12 and 46. The rejection of these claims should be withdrawn.

The Examiner rejected claims 17 and 51 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Anshel et al.

The Examiner stated that Ming et al., at column 4, lines 18-20, describe modifying the divisor of static data. However, a close reading of this passage does not reveal the subject matter described by the Examiner. Rather, the passage sets forth a procedure for updating seed values for generating the pseudorandom sequence, based on an incremented frame count. Anshel et al. disclose a cryptographic sequence generator. In the passage cited by the Examiner, Anshel et al. describe use of randomly-selected prime numbers to generate a list of Jacobi symbols and, according to a public key, chooses a subset of the symbols to encrypt a single input bit. The Jacobi symbols and the public key both feature terms having prime number divisors. See column 11, lines 8-53. In contrast, claims 17 and 51 recite updating the static data by modifying a prime number divisor of the static data. The static data is a basis for a random key split.

None of the cited references discloses random key splits. It follows that none of the cited references discloses updating a basis of a random key split, or performing the update by modifying a prime number divisor of the basis. Anshel et al. disclose the use of prime number divisors in a Jacobi sequence and in a public key. However, demonstrating that it is known to use prime number divisors in an application related to cryptography is not enough to suggest to one of ordinary skill in the art that modification of a prime number divisor of a basis of a random key split is beneficial in generating a key based on the randomization of a number of key splits, particularly in view of other references that do not even disclose the use of key splits. It is clear that no combination of the teachings of the cited references could possibly render obvious the invention recited in claims 17 and 51. The rejection of these claims, therefore, should be withdrawn.

The Examiner rejected claims 18, 20-24, 52, and 54-58 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Ming et al., and further in view of Anshel et al.

Regarding claims 18 and 52, the Examiner stated that Anshel et al. discuss a console split generator for generating a console key split based on maintenance data, citing column 8, lines 8-15. The passage cited by the Examiner actually describes a conventional public key infrastructure arrangement. As disclosed, users of a network are each given a <u>fixed</u> (line 10) private key, and a public key is broadcast to users of the network. Key splits of any kind are not disclosed. As disclosed, the private key is fixed; this would teach away from having a key component that would depend on maintenance data. In the Anshel et al. scheme, this private key remains fixed, while each public key is used only once and then discarded. A new public key is generated, based solely on an incremented state value. There is no seed provided to generate a split from maintenance data, or randomizer for combining a console split with other splits to determine a key, as recited in claims 18 and 52. Rather, the key is determined directly from the state data.

As previously discussed, Hirsch and Ming et al. do not disclose the elements of claims 1 and 35, from which claims 18 and 52 respectively depend. The Examiner acknowledged that these references also do not disclose the particular features of claims 18 and 52. Thus, no combination of the teachings of these references could render obvious the invention recited in claims 18 and 52. The rejection of these claims should be withdrawn.

Regarding claims 20 and 54, the Examiner stated that Anshel et al. disclose means for generating a pseudorandom sequence based on maintenance data, citing column 8, lines 8-15. As described in the passage and shown in Figure 4, the ZPNG generates a pseudorandom code which is transformed by a zeta code transformer to produce the public key. This, therefore, has nothing to do with a split generator that generates one of a number of key splits that are randomized to form a cryptographic key. Rather, the sequence itself is transformed to become the key. As previously discussed, Hirsch and Ming et al. do not disclose such a combiner, so it would be pointless to apply the teachings of Anshel et al. to those of these two references to show that the Anshel et al. key could be a key split in a Hirsch/Ming et al. system. Thus, no combination of the teachings of these references could render obvious the invention recited in claims 20 and 54. The rejection of these claims should be withdrawn.

Regarding claims 21 and 55, the Examiner stated that Anshel et al. specify generating a key split based on previous and current maintenance data, citing column 8, lines 26 and 27. As described in the passage, a public key is generated based on a current state value. The key is used once, and discarded. The state value is incremented, and a new public key is determined based on this incremented value. Thus, the public key is determined based only on a current state value. The previous state value and current state value are never both used to determine the public key, as required by claims 21 and 55. See also Figure 4. Further, this key is not a key split, as discussed previously. Also as previously discussed, Hirsch and Ming et al. do not disclose the claimed combiner, and the Examiner acknowledged that these references do not disclose the particular features recited in claims 21 and 55. Thus, no combination of the teachings of these references could render obvious the invention recited in claims 21 and 55. The rejection of these claims should be withdrawn.

Regarding claims 22 and 56, the Examiner stated that Anshel et al. mention generating a key split based on maintenance data and static data, citing column 8, lines 16-22. The passage cited by the Examiner actually describes generation of the public key of a conventional public key infrastructure arrangement. As disclosed, the public key is generated based on an incremented state value. There is no seed provided to generate a split from maintenance data, or randomizer for combining a console split with other splits to determine a key, as recited in claims 22 and 56. Rather, the key is determined directly from the state data.

As previously discussed, Hirsch and Ming et al. do not disclose the elements of claims 1 and 35, from which claims 22 and 56 respectively depend. That is, like Anshel et al., Hirsch and Ming et al. do not disclose a number of key splits, generated from different seeds, that are randomized to generate a key. The Examiner acknowledged that these references also do not disclose the particular features of claims 22 and 56. Thus, no combination of the teachings of these references could render obvious the invention recited in claims 22 and 56. The rejection of these claims should be withdrawn.

Regarding claims 23 and 57, the Examiner stated that Anshel et al. delineate a means for updating static data, citing column 8, lines 8, 26, and 27. This passage describes generating a new public key based on an incremented state value. It is believed that the Examiner previously identified the disclosed state data with the claimed maintenance data. Claims 23 and 57, through their respective dependence from claims 22 and 56, require both maintenance data and static data. It is not clear how these individual elements are delineated in the Anshel et al. invention. That is, a close reading of the disclosure of the invention, and of the cited passage in particular, does not reveal a correspondence of disclosed elements with claimed elements. The lack of detail in the Examiner's assertions does not assist in making this identification. In any case, Anshel et al. does not disclose both maintenance and static data, which is updated, as components of a key split that is randomized with other key splits to produce a key, as recited in the claims. As previously discussed, the other cited references also fail to disclose these elements of the claimed invention. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 23 and 57. The rejection should be withdrawn.

Regarding claims 24 and 58, the Examiner stated that Anshel et al. illustrate that updating the static data includes modifying a prime number divisor of the static data, citing column 11, lines 8-25 and Figure 8, item 71. In the passage cited by the Examiner, Anshel et al. describe use of randomly-selected prime numbers to generate a list of Jacobi symbols and, according to a public key, chooses a subset of the symbols to encrypt a single input bit. The Jacobi symbols and the public key both feature terms having prime number divisors. See column 11, lines 8-53. Further, the described process in not connected in any way to anything that the Examiner has identified as being static data, or to the updating of this static data. In contrast, claims 24 and 58 recite updating the static data by modifying a prime number divisor of the static data. Demonstrating that it is known to use prime number divisors in an application related to cryptography is not enough to suggest to one of ordinary skill in the art that modification of a prime number divisor of static data basis of a key split is beneficial in generating a key based on the randomization of a number of key splits, particularly in view of other references that do not even disclose the use of key splits. It is clear that no combination of the teachings of the cited references could possibly render obvious the invention recited in claims 24 and 58. The rejection of these claims, therefore, should be withdrawn.

The Examiner rejected claims 19 and 53 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Anshel et al., and further in view of Albert et al. The Examiner stated that Anshel et al. describe means for generating a pseudorandom sequence based on maintenance data (column 8, lines 8-15), and that Albert et al. specify a random sequence (column 1, line 66 - column 2, line 2).

Claims 19 and 53 require generating a console key split, among other key splits, that are to be randomized to produce a key. The console key split is based on maintenance data, and the console split generator (or generating the console split) includes generation of a random sequence based on the maintenance data. As previously discussed, Hirsch does not disclose any of these elements. Anshel et al. describe generating a public key from a pseudorandom sequence based on a state value. Anshel et al. do not disclose a key generated by randomizing a number of key splits. Albert et al. merely describe a circuit for generating a random sequence. Thus, none of the references discloses or suggests all the elements of claims 19 and 53, that is, generating a console key split, among other key splits, that are to be randomized to produce a key, where the console key split is based on maintenance data, and the console split generator (or generating the console split) includes generation of a random sequence based on the maintenance data. Therefore, no combination of the teachings of the cited references could render obvious the invention recited in claims 19 and 53. The rejection should be withdrawn.

The Examiner rejected claims 25, 27-31, 59, and 61-65 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Tomko et al.

Regarding claims 25 and 59, the Examiner stated that Tomko et al. elaborate on a biometric split generator for generating a biometric key split based on biometric data. Tomko et al. disclose a fingerprint controlled public key cryptographic system. Tomko et al. utilize data derived from a user's fingerprint to generate a private key to be used in encrypting and decrypting messages. The fingerprint data is provided as an input to a pseudorandom sequence generator to generate the key. No other components are included in the key, and therefore the biometric data used in the Tomko et al. system is not used to generate a biometric key split, as recited in the claims, but rather is used to generate the key itself. No other component is randomized with the biometric data to derive the key. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35.

Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 25 and 59. The rejection of these claims should be withdrawn.

Regarding claims 27 and 61, the Examiner stated that Tomko et al. disclose means for generating a pseudorandom sequence based on the biometric data. However, claims 27 and 61 require biometric split generation, which, as noted above in discussing the rejection of claims 25 and 59, is not disclosed by Tomko et al. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35, from which claims 27 and 61 depend. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 27 and 61. The rejection of these claims should be withdrawn.

Regarding claims 28 and 62, the Examiner stated that Tomko et al. delineate means for generating a key split based on biometric data vectors and on biometric combiner data. However, Tomko et al. do not disclose generating a key split at all. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35, from which claims 28 and 62 depend. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 28 and 62. The rejection of these claims should be withdrawn.

Regarding claims 29 and 63, the Examiner stated that Tomko et al. explain a means for generating a key split based on biometric data and on static data. However, Tomko et al. do not disclose generating a key split at all. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35, from which claims 29 and 63 depend. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 29 and 63. The rejection of these claims should be withdrawn.

Regarding claims 30 and 64, the Examiner stated that Tomko et al. illustrate updating the static data. However, Tomko et al. do not disclose generating a biometric key split, or any key split at all, as recited in the claims. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35, from which claims 30 and 64 depend. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 30 and 64. The rejection of these claims should be withdrawn.

Regarding claims 31 and 65, the Examiner stated that Tomko et al. elaborate that the means for updating the static data includes means for modifying the prime number divisor of the static data, citing column 7, line 45 - column 8, line 12. However, this passage describes the procedure used to derive an array *b* that is related to the unique number *u*, which is derived from the Fourier transform of the user's fingerprint data. That is, the passage describes the modular mathematics used to derive the coefficients *b* that determine *u*. Prime numbers are not mentioned at all, and certainly not in terms of modifying the prime number divisor of any static data. No primes are used, and the only data used is derived from the biometric data.

Further, Tomko et al. do not disclose generating a key split at all. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35, from which claims 31 and 65 depend. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 31 and 65. The rejection of these claims should be withdrawn.

The Examiner rejected claims 26 and 60 under 35 USC §103(a) as being unpatentable over Hirsch, in view of Tomko et al., and further in view of Albert et al. Claims 26 and 60 recite generating (and means for generating) a random sequence based on biometric data for biometric split generation. The Examiner stated that Tomko et al. discuss means for generating a pseudorandom sequence based on biometric data, and that Albert et al. specify a random sequence.

However, Tomko et al. do not disclose generating a key split. Rather, Tomko et al. disclose generating a key based solely on biometric data, not on randomized splits. Albert et al. merely describe a random sequence generator. As discussed above, Hirsch does not disclose a randomizer for combining key splits derived from individual seeds to generate a key, as recited in claims 1 and 35, from which claims 26 and 60 depend. Thus, no combination of the teachings of the cited references could render obvious the invention recited in claims 26 and 60. The rejection of these claims should be withdrawn.

Based on the foregoing, it is submitted that all rejections have been successfully traversed. It is therefore requested that the claims be allowed and the case passed to issue.

A petition for extension of time is filed herewith. A check in payment of the fee for the extension is also enclosed. If the check is missing or made out for an insufficient amount, please charge our deposit account, No. 18-0002, and notify us accordingly.

Respectfully submitted,

February 18, 2000
Date

Thomas M. Champagne
Registration No. 36,478
RABIN & CHAMPAGNE, P.C.
1725 K Street, N.W., Suite 1111
Washington, D.C. 20006
(202) 659-1915
(202) 659-1898 fax

TMC:lep

Certification under 37 CFR 1.8

February 18, 2000
Date of Deposit

I hereby certify that this Amendment, along with a proposed drawing change, a petition for extension of time, and check in payment of the petition fee, is being deposited with the United States Postal Service as First Class Mail under 37 CFR §1.8 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Thomas M. Champagne
Typed or printed name of person mailing Amendment

Signature of person mailing Amendment